Http 与 cookie

1.Http

http（超文本传输协议）是一个基于请求与响应的应用层协议。

url请求资源的地址  http://host[":"port][abs\_path]

host 主机名，对应IP地址的一个点或一段；port 端口号 ；abs\_path 主机上的资源路径

请求方式：

  get请求，将请求数据作为url一部分发送，不安全，传输数据量小，方便易用。

  post请求，传输数据量大，安全，一般做表单提交。

常见响应状态码：

200 OK        //客户端请求成功

400 Bad Request  //客户端请求有语法错误，不能被服务器所理解

401 Unauthorized //请求未经授权，这个状态代码必须和WWW-Authenticate报头域一起使用

403 Forbidden  //服务器收到请求，但是拒绝提供服务

404 Not Found  //请求资源不存在，输入了错误的URL

500 Internal Server Error //服务器发生不可预期的错误

503 Server Unavailable  //服务器当前不能处理客户端的请求，一段时间后可能恢复正常

2.cookie

什么是cookie:会话跟踪技术

什么是会话：用户进入网站  开始浏览到结束浏览的这样的一个过程 就称为一次会话

会话跟踪技术: 浏览器（B端）和服务器 之间在进行多次请求数据时，数据共享的过程 就称为 会话跟踪技术

客户端： 为用户提供上网的一台pc机

服务器： 为客户端提供服务

cookie的生成过程： 打开浏览器--->输入网址--->回车--->域名解析--->访问服务器

                  --->返回主页--->将cookie存储在浏览器

                  用户要访问二级页面时，打开新的窗口，会自动判断上一个页面中

                  是否含有cookie,如果有，就将cookie数据带入到下一个页面

cookie 是存储于访问者的计算机中的变量。每当同一台计算机通过浏览器请求某个页面时，就会发送这个 cookie。你可以使用 JavaScript 来创建和取回 cookie 的值。

你可以把用户的登录名密码存储于cookie中，就可以实现自动登录。也可以用来保存购物车信息。

3.cookie数据的存和取

document.cookie = "键=值" 存

document.cookie           取

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8" />

        <title></title>

    </head>

    <body>

        <input type="button" value="存" id = "s"/>

        <input type="button" value="取" id = "g" />

        <a href="2.html" target="\_blank">跳转至二级页面</a>

    </body>

</html>

<script>

    let oSave = document.getElementById("s");

    let oGet = document.getElementById("g");

    //存cookie

    oSave.onclick = function(){

        document.cookie = "name=laowang";

        alert("存储cookie成功");

    }

    oGet.onclick = function(){

        alert(document.cookie);

    }

</script>

多个cookie的存取

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

    </head>

    <body>

        <input type="button" value="存" id="s"/>

        <input type="button" value="取" id="g"/>

    </body>

</html>

<script>

    let oSave = document.getElementById("s");

    let oGet= document.getElementById("g");

    oSave.onclick = function(){

        document.cookie = "name=laowang";

        document.cookie = "pwd=123456";

        //注意多个cookie存储时,使用分号+空格,千万不要忘了空格

        console.log(document.cookie);

    }

    oGet.onclick = function(){

        let strCookie = document.cookie;

        //注意这一步贼关键分号+空格

        let arrCookie = strCookie.split("; ");

        for(let t in arrCookie){

            //console.log(arrCookie[t]);

            let item = arrCookie[t].split("=");

            console.log("姓名"+item[0],"密码"+item[1]);

        }

    }

</script>

cookie实现注册和登录

注册:

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

    </head>

    <body>

        用户名:<input type="text" id="t1" /></br>

        密码:<input type="text" id="t2" /></br>

        <input type="button" value="注册" id="btn"/>

    </body>

</html>

<script>

    let name = document.getElementById("t1");

    let pwd = document.getElementById("t2");

    let btn = document.getElementById("btn");

    btn.onclick = function(){

        document.cookie = "name="+t1.value;

        document.cookie = "pwd="+t2.value;

        location.href = "5.html";

    }

</script>

登录:

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

        用户名:<input type="text" id="t1" /></br>

        密码:<input type="text" id="t2" /></br>

        <input type="button" value="登录" id="btn"/>

    </head>

    <body>

    </body>

</html>

<script>

    //页面加载获取cookie数据

    window.onload = function(){

        //如果用户输入的用户名密码 == cookie的用户名和密码则登录成功

        let name = document.getElementById("t1");

        let pwd = document.getElementById("t2");

        let btn = document.getElementById("btn");

        let strCookie = document.cookie;

        let arrCookie = strCookie.split("; ");

        let myName="";

        let myPwd="";

        for(let i=0; i<arrCookie.length; i++){

            let item = arrCookie[i].split("=");

            //循环寻找值

            if(item[0] == "name"){

                myName = item[1];

            }

            if(item[0] == "pwd"){

                myPwd = item[1];

            }

        }

        btn.onclick = function(){

            if(myName == name.value && myPwd == pwd.value){

                alert("登录成功");

            }else{

                alert("用户名或密码错误");

            }

        }

    }

</script>

切记存储cookie时，都以字符串来存取

JSON.parse()【字符串转为json对象，从一个字符串中解析出JSON对象】

JSON.stringify()【json对象转为字符串，从一个JSON对象中解析出字符串】

4.cookie的生存周期

生存周期  就是cookie数据在浏览器上保存的时间

document.cookie = "键=值;expires=标准时间格式"

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

    </head>

    <body>

        用户名:<input type="text" id="t1" /></br>

        密码:<input type="text" id="t2" /></br>

        <input type="button" value="注册" id="btn"/>

    </body>

</html>

<script>

    let name = document.getElementById("t1");

    let pwd = document.getElementById("t2");

    let btn = document.getElementById("btn");

    btn.onclick = function(){

        let date = new Date();

        date.setDate(date.getDate()+11);

        //expires设置有效时间，必须为标准时间date.toGMTString();

                                        //"name=xxx;expires=xxx"

        document.cookie = "name="+t1.value+";expires="+date;//date.toGMTString();

         document.cookie = "pwd="+t2.value+";expires="+date;

        //location.href = "5.html";

    }

</script>

5.cookie的删除

cookie不能直接删，需要侧面删除

两种方法:

    a.将键对应的值设为空

    b.expires="-1";

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

    </head>

    <body>

    </body>

</html>

<script>

    let d = new Date();

    d.setDate(d.getDate()+2);

    //document.cookie="user=dahuang;expires="+d;

    document.cookie="user='';expires=-1";

</script>

6.cookie特点

Cookie的特点:

（1）cookie可能被禁用。

（2）cookie是与浏览器相关的。不同浏览器所保存的cookie也是不能互相访问的。

（3）cookie可能被用户删除。

（4）cookie安全性不够高。如果要保存用户名密码等信息时，最好事先经过加密处理。

（5）存储的数据量 4k 大小，cookie只支持存储string类型的数据。

（6）简单易用。

（7）信息存储于用户硬盘，因此可以作为全局变量。

7.cookie的封装

<script>

    //封装一个获取cookie值得方法

    function getCookie(key){

        let str = document.cookie;

        let arr = str.split("; ");//分号后的空格不要删 cookie之间的键值对是通过分号+空格

        for(let i=0; i<arr.length; i++){

            let item = arr[i].split("=");

            if(item[0] == key){

                return item[1];

            }

        }

        //没cookie，或者有cookie,但是没找到key 得不到cookie, 返回一个空字符串

        return ";"

    }

    //封装一个设置cookie值得方法

    function setCookie(key,value,day){

        let d = new Date();

        d.setDate(d.getDate()+day);

        document.cookie = key+"="+value+";expires=" + d;

    }

    //删除cookie

    function delCookie(key){

        setCookie(key," ",-1);

    }

    let name = getCookie("name");

    console.log(name);

    setCookie("name","小明",10);

    name = getCookie("name");

    console.log(name);

    delCookie("name");

</script>